Remarks

The applicants have carefully considered the Office action dated October 4, 2006. In view of the forgoing amendments and the following remarks, reconsideration of the application is respectfully requested.

In the Office action, claims 1-4 and 10-19 were rejected as anticipated by Ju (US 6,260,190), claims 5-7 and 20-22 were rejected as unpatentable over Ju in view of McKinsey (US 6,463,579) and Babaian (US 7,065,750), and claims 8, 9, 23, and 24 were rejected as unpatentable over Ju in view of McKinsey.

Claim 1 recites a method comprising, *inter alia*, determining that a software instruction is an excepting instruction and inserting a control speculative version of the software instruction at a second location within the plurality of software instructions in response to determining that the software instruction is an excepting instruction.

The portions of Ju cited in the Office action do not describe or suggest inserting a speculative version of a software instruction in response to determining that that a software instruction is an excepting instruction. In direct opposition to claim 1, Ju describes inserting an instruction regardless of whether an instruction is an excepting instruction.

Ju is directed to a unified compiler framework for control and data speculation with recovery code. Ju describes inserting an eager mode instruction before an original location of an instruction in a set of code. As per Ju, if the eager mode instruction does not generate an exception during execution, the eager mode instruction executes normally. If the eager mode instruction generates an exception during execution, the exception is logged and a speculative tag is set in a register. (Col. 23, lines 9-38). In other words, even if the eager mode instruction described by Ju is a speculative instruction, a point which the applicants do not concede, Ju does not describe or suggest that the eager mode instruction is inserted in

response to determining that the software instruction is an excepting instruction. In fact, as noted above, Ju inserts the instruction prior to determining that the instruction has generated an exception because Ju is directed to determining if the inserted instruction has generated an exception during execution, in direct contrast to determining that the replaced instruction is capable of generating an exception. (Col. 23, lines 9-13 precede Col. 23, lines 19-23 and Col. 23, lines 32-37). Accordingly, Ju cannot not describe inserting the instruction in response to the determination, prior to making such a determination. Therefore, because Ju fails to describe or suggest all of the elements of claim 1, claim 1 and all claims depending therefrom are in condition for allowance.

Independent, claim 5 recites a method comprising, *inter alia,* receiving a plurality of instructions including a first instruction, determining if the first instruction is an excepting instruction, determining if the first instruction is to be moved upward across a check instruction, and determining a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction when the first instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction. The Office action contended that "Ju describes moving a first instruction that is not an excepting instruction upward across a check instruction ... [Ju] shows moving the first add instruction upward across a check instruction, and ... shows that the load instruction is the excepting instruction." (Page 6, lines 3-6). However, while the add instruction may not be described as generating an exception during the particular execution example described in Ju, an add instruction is inherently an excepting instruction. For example, an add instruction is capable of triggering an arithmetic overflow exception. An instruction that is capable of triggering an arithmetic overflow exception is an excepting instruction. (see, for example, ¶ [0025] of the present application). Therefore, Ju does not

describe or suggest receiving a plurality of instructions including a first instruction, determining if the first instruction is an excepting instruction, determining if the first instruction is to be moved upward across a check instruction, and determining a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction when the first instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction.

Similar to Ju, the cited portions of McKinsey (US 6,463,579) do not describe or suggest determining that a first instruction is an excepting instruction, nor does the Office action cite McKinsey for this purpose. Therefore, the cited portions of McKinsey cannot describe or suggest determining a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction when the first instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction.

Likewise, the cited portion of Babian et al. (US 7,065,750) do not describe or suggest determining a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction when the first instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction. Rather, Babian describes saving the contents of registers in a temporary location before executing an instruction that could generate an exception. (Col. 7, lines 56-62). Further, the cited portion of Babian does not suggest the recitations of claim 5 because the cited portion is directed to handling exceptions that are triggered during execution of instructions, whereas claim 5 recites determining a second instruction when the first instruction is not an excepting instruction.

Because Ju, McKinsey, and Babian fail to describe or suggest receiving a plurality of instructions including a first instruction, determining if the first instruction is an excepting instruction, determining if the first instruction is to be moved upward across a check instruction, and determining a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction when the first instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction as recited in claim 5, no combination of Ju, McKinsey, and Babian can describe or suggest such recitations. Therefore, claim 5 and all claims depending therefrom are in condition for allowance.

In the Office action, claim 8 was rejected as unpatentable over Ju in view of McKinsey. Claim 8 recites a method comprising receiving a plurality of instructions including a first instruction, determining if the first instruction is an excepting instruction, determining if the first instruction is to be moved upward across a check instruction, determining if the first instruction is to be moved downward across the check instruction, and inserting a copy of the first instruction into a recovery block if (i) the first instruction is not an excepting instruction, (ii) the first instruction is not to be moved upward across a check instruction, and (iii) the first instruction is to be moved downward across a check instruction. As described above in connection with claim 1, Ju fails to describe or suggest inserting an instruction in response to determining that a first instruction is not an excepting instruction.

Similar to Ju, as described above in connection with claim 5, McKinsey does not describe or suggest determining that a first instruction is not an excepting instruction. Therefore, McKinsey cannot describe or suggest inserting a copy of the first instruction into a recovery block in response to determining that the first instruction is not an excepting instruction.

For at least the forgoing reasons, Ju and McKinsey fail to describe or suggest all of the recitations of claim 8. Accordingly, no combination of Ju and McKinsey can describe or suggest claim 8. Therefore, claim 8 and all claims depending therefrom are in condition for allowance.

In the Office action, claim 10 was rejected as anticipated by Ju. Claim 10 recites an apparatus comprising, *inter alia*, a control speculation module operatively coupled to the processor, the control speculation module being structured to insert a control speculative version of a software instruction into the plurality of software instructions in response to a determination that the software instruction is an excepting instruction. As described above in connection with claim 1, Ju does not describe or suggest inserting a control speculative version of a software instruction in response to a determination that the software instruction is an excepting instruction. Therefore, for at least the forgoing reasons, claim 10 and all claims depending therefrom are in condition for allowance.

In the Office action, claim 13 was rejected as anticipated by Ju. Claim 13 recites an apparatus comprising a computer device structured to, *inter alia*, determine that the software instruction is an excepting instruction and insert a control speculative version of the software instruction at a second location within the plurality of software instructions in response to determining that the software instruction is an excepting instruction. As described above in connection with claim 1, Ju does not describe or suggest inserting a control speculative version of a software instruction in response to a determination that the software instruction is an excepting instruction. Therefore, for at least the forgoing reasons, claim 13 and all claims depending therefrom are in condition for allowance.

In the Office action, claim 17 was rejected as anticipated by Ju. Claim 17 recites an apparatus comprising a computer device structured to, *inter alia*, determine that the software

instruction is an excepting instruction and insert a control speculative version of the software instruction at a second location within the plurality of software instructions in response to determining that the software instruction is an excepting instruction. As described above in connection with claim 1, Ju does not describe or suggest inserting a control speculative version of a software instruction in response to a determination that the software instruction is an excepting instruction. Therefore, for at least the forgoing reasons, claim 17 and all claims depending therefrom are in condition for allowance.

In the Office action, claim 20 was rejected as unpatentable over Ju in view of McKinsey and Babian. Claim 20 recites a machine readable medium structured to cause a machine to, *inter alia*, receive a plurality of instructions including a first instruction, determine if the first instruction is an excepting instruction, determine if the first instruction is to be moved upward across a check instruction, determine a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction when the first instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction, determine if a source operand associated with the second instruction is available at the check instruction, insert a third instruction into the plurality of instructions to save the value of the target register if the source operand associated with the second instruction is not available at the check instruction, and insert a fourth instruction into a recovery block to restore the value of the target register. As described above in connection with claim 5, no combination of Ju, McKinsey, and Babian can describe or suggest determining a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction if the first instruction is not an excepting instruction. Therefore, for at least the forgoing reasons, claim 20 and all claims depending therefrom are in condition for allowance.

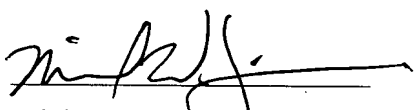
In the Office action, claim 23 was rejected as unpatentable over Ju in view of McKinsey. Claim 23 recites a machine readable medium structured to cause a machine to receive a plurality of instructions including a first instruction, determine if the first instruction is an excepting instruction, determine if the first instruction is to be moved upward across a check instruction, determine if the first instruction is to be moved downward across the check instruction, and insert a copy of the first instruction into a recovery block in response to determining that (i) the first instruction is not an excepting instruction, (ii) the first instruction is not to be moved upward across a check instruction, and (iii) the first instruction is to be moved downward across a check instruction. As described above in connection with claim 8, no combination of Ju and McKinsey can describe or suggest inserting a copy of the first instruction into a recovery block in response to determining that the first instruction is not an excepting instruction. Therefore, for at least the forgoing reasons, claim 23 and all claims depending therefrom are in condition for allowance.

Reconsideration of the application and allowance thereof are respectfully requested. If there is any matter that the examiner would like to discuss, the examiner is invited to contact the undersigned representative at the telephone number set forth below.

Respectfully submitted,

HANLEY, FLIGHT & ZIMMERMAN, LLC
150 S. Wacker Dr.
Suite 2100
Chicago, Illinois 60606

Dated: **March 5, 2007**



Michael W. Zimmerman
Reg. No. 57,993
Agent for Applicants
312.580.1020